

# Arduino と Android と Wiki を使ったセンサネットワークシステム

山之上卓<sup>†1</sup> 小田謙太郎<sup>†1</sup> 下園幸一<sup>†1</sup>

Android と Arduino を組み合わせて構成された遠隔端末と、Wiki サーバで構成されたセンサネットワークシステムについて述べる。遠隔端末はオープンソースハードウェアのマイコンボードの1つである Arduino にセンサとアクチュエータを接続したものと、Android 端末で構成されている。Android を使うことにより、Android 端末がインターネットに接続できるのであれば、LAN や無線 LAN が利用できない場所のデータも、インターネット上の Web サーバ経由で入手可能となる。この Web サーバを経由して、Arduino に接続されたアクチュエータを制御することも可能になる。一般的な Wiki ソフトウェアを利用することにより、特定の Web サイトや Web ソフトウェアに縛られることなく、広くセンサネットワークシステムを利用することが可能になる。人間同士だけでなく、人間と遠隔端末間の情報共有を容易にすることも可能になる。

## A Sensor Network System which adopts Arduino, Android and Wiki

TAKASHI YAMANOUÉ<sup>†1</sup> KENTARO ODA<sup>†1</sup> KOICHI SHIMOZONO<sup>†1</sup>

A sensor network system, which consists of remote terminals and Wiki servers, is discussed. A remote terminal consists of an Arduino board, a microcomputer board of an open source hardware, with sensors and actuators, and an Android terminal. Data at an every remote place, where the Android terminal can access the Internet even if a LAN or Wi-fi connection can't available, can be available through a Wiki server on the Internet. Actuators at the remote place also can be controlled from another place using the Web server. This sensor network system can be used freely without specifying web sites or web services because commonly available wiki software is used for the Web server. This system enables sharing information among people and machines not only among only people.

### 1. はじめに

Wiki[17]はブラウザを使って、簡単に相互にリンクされた Web ページを作ったり、その Web ページを編集したりすることができる Web サイトであり、協同作業したり、情報の共有をしたりすることを効果的に行うためによく利用されている[16].

もし、Wiki が人間にとって優しいものであるのなら、Wiki は機械にとっても優しいものであるはずである。もし機械が自動的に Wiki ページのデータを読んだり書いたりすることができるなら、人間はより多くの有用な情報を得ることができるはずである。また、Wiki ページを通して、人間が簡単に機械を制御することも可能になるはずである。人間-機械間の通信だけでなく、機械間の通信も可能になるはずである。これらのことにより、機械が Wiki のページを読み書きできるようになることにより、Wiki は従来の人間だけが使うものより有益なものになるはずである。例えば、よく利用されている Wiki がセンサネットワークのセンサを接続するために利用できるようなれば、我々自身のセンサネットワークを簡単に構築できるようになるはずである。

これらのことを確認するため、我々は Wiki を使ったセンサネットワークシステムを開発している。このセンサネットワークシステムは、Android[8]と Arduino[9]を組み合わせて構成された遠隔端末と、Wiki サーバで構成されている。

遠隔端末はオープンソースハードウェアのマイコンボードの1つである Arduino にセンサとアクチュエータを接続したものと、Android 端末で構成されている。Android を使うことにより、Android 端末がインターネットに接続できるのであれば、LAN や無線 LAN が利用できない場所のデータも、インターネット上の Web サーバ経由で入手可能となる。この Web サーバを経由して、Arduino に接続されたアクチュエータを制御することも可能になる。一般的な Wiki ソフトウェアを利用することにより、特定の Web サイトや Web ソフトウェアに縛られることなく、広くセンサネットワークシステムを利用することが可能になる。

PukiWiki[13]は日本で一般的に使われている Wiki ソフトウェアの一つである。筆者らは PukiWiki のページを読み書き可能にし、なおかつ、PukiWiki のページから起動可能な Java Applet の API を開発している[2][3][4][5]。この API を、PukiWiki-Java Connector (PJC) と呼んでいる。PJC を改造することにより、Android 端末で PukiWiki のページを読み書きを可能にする API、PukiWiki-Java Connector Service for Android (PJC-S) を作成した。これを本センサネットワークシステムにおける、Android 端末 - Wiki (PukiWiki)サーバ間の通信に利用する。

本センサネットワークシステムの遠隔端末の中で、Arduino と Android 端末間の接続には、Android Open Accessory Development Kit (ADK)[9]を利用した。

本センサネットワークシステムを使って、遠隔地の人間の活動状況を把握するシステムを作成することができた。

<sup>†1</sup> 鹿児島大学  
Kagoshima University

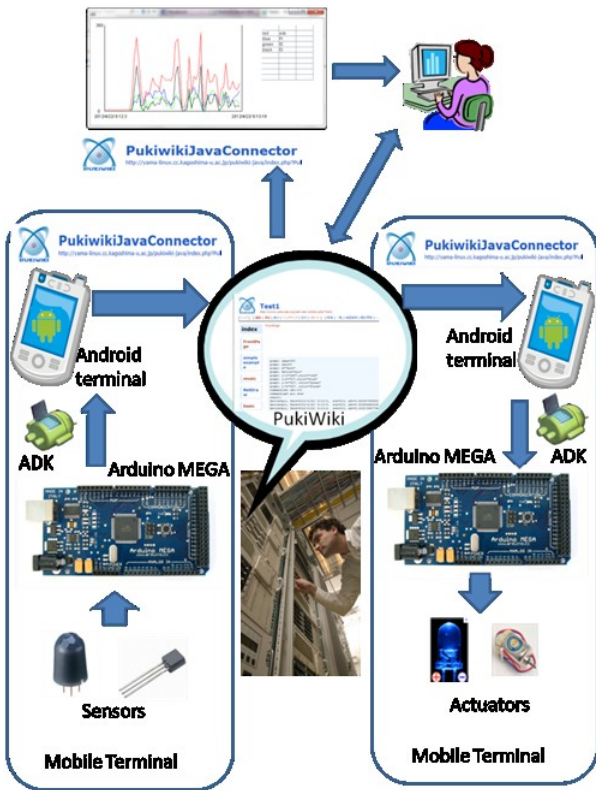


図 1. 本システムの概要

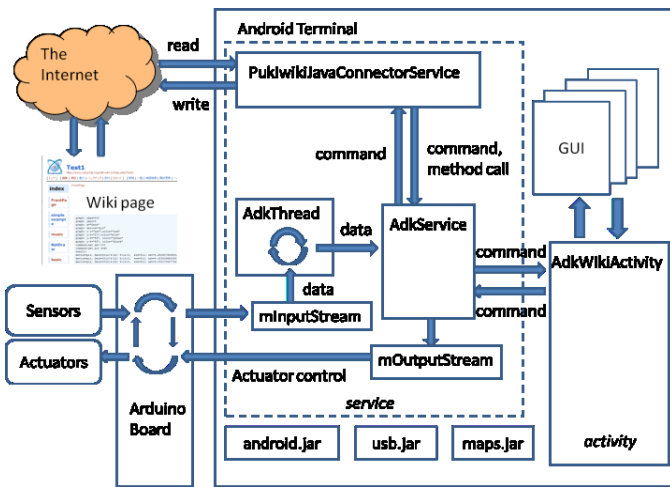


図 2. 遠隔端末の詳細

## 2. システム概要

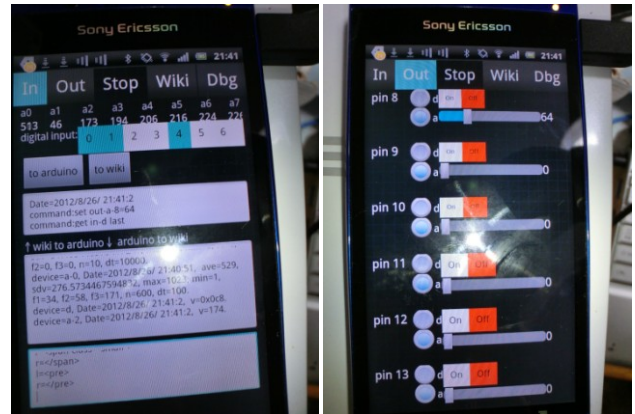
図 1 に本システムの概略を示す. 端末システム(Mobile Terminal)は, 物理的には, Android と Arduino と Arduino に接続されたセンサやアクチュエータで構成されている.

Android 端末はセンサから入力されたデータを処理し, PJC-S を使って, PukiWiki サーバの wiki ページにそれを書き込む. Wiki ページに書かれたコマンドは PJC-S を使って Android 端末に読み込まれる. Android 端末はこのコマンドを処理し, ADK を使って Arduino のアクチュエータ

を操作する. 遠隔端末によって入手された Wiki ページのデータは PukiWiki サイトの PJC によって処理することも可能である. これにより, データはグラフなどで視覚化することができる.

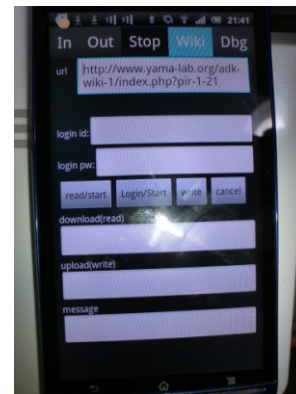
## 3. 遠隔システムの詳細

図 2 に遠隔端末の詳細を示す. Arduino と Android 端末は USB ケーブルで接続される. Android 端末の中で, 本システムのアプリケーションプログラムが動作する. このアプリケーションプログラムは AdkService, PukiWikiJavaConnectorService (PJC-S), AdkThread, AdkWikiActivity などの部品によって構成されている. Android.jar, usb.jar, maps.jar などが USB 通信のため利用される. 図 3 に端末システムの Android 端末の表示画面の例を示す.



(a) 入力データ表示画面

(b) 出力制御画面



(c) Wiki 接続画面

図 3. Android 端末の GUI

### 3.1 AdkService

AdkService は遠隔端末の中心の部品である. これは service プログラムであり, Android 上のバックグラウンドプロセスとして動作する. AdkService は AdkActivity によって起動される. AdkService は AdkWikiActivity から停止メッセージが送られ, これを受信するまで, 動作を継続する. AdkService は AdkWikiActivity が停止メッセージを送

らずに停止しても停止しない。この性質は Android で長時間動作するシステムを作成するのに必須のものである。

AdkService は handler のメッセージパッシングを使って AdkWikiActivity と通信を行う。AdkService は以下のようなメッセージを AdkWikiActivity から受け取る。

- USB コントローラにアクセスするためのファイル記述子。Arduino にアクセスする I/O ストリームは `mInputStream` と `mOutputStream` として、このファイル記述子から得られる。
- AdkService から AdkWikiActivity へメッセージを送るための `callback` ポインタ
- 停止コマンドのような、AdkService を制御するコマンド
- PJC-S を制御するコマンド。

AdkService は AdkWikiActivity に対し、GUI を制御するなどのコマンドを送信する。

AdkService は PJC-S との間で、メッセージパッシングとメソッド呼び出しによって通信を行い、Arduino との間では `mInputStream` と `AdkThread` を使ってデータを読み込み `mOutputStream` をつかって Arduino にアクチュエータの制御データを送る。AdkService が Arduino から読み込んだデータは PJC-S に送られ、同じものが表示用として AdkWikiActivity にも送られる。

### 3.2 PukiwikiJavaConnectorService

PukiwikiJavaConnectorService (PJC-S) は PukiWiki-Java を改造して作成した Android の部品である。この部品は AdkService からコマンドを読んで、それを解釈実行する。コマンドは PukiWiki サーバと通信するために使われたり、AdkService と通信するために使われる。この部品を構成する class は以下の public method を含む。

```
public PukiWikiConnectorService(  
    PukiwikiJavaApplication service)
```

このコンストラクタにおいて、`service` は、この PJC-S と AdkService が通信するために使われる。

```
public boolean parseCommand(  
    String command, String value)
```

このメソッドは、PJC-S が外部からコマンドを受け取るために利用される。コマンドには Wiki ページの URL を指定したり、認証サイトへのログインを行ったり、指定された URL のページを読み込んだり、指定された URL のページに書き込みを行ったりするものがある。

PJC-S は AdkService の `method` を呼び出すことができる。移植性を持たせるため、AdkService は PukiwikiJavaAppication インターフェースを実装したものになっている。このインターフェースの定義を以下に示す。

```
public interface PukiwikiJavaApplication {  
    public String getOutput();  
    public void setInput(String x);  
    public void setSaveButtonDebugFrame(  
        SaveButtonDebugFrame f);  
    public void sendCommand(String c, String v);  
}
```

`getOutput` メソッドは、PJC-S が PukiWikiJavaApplication (この場合は AdkService) からテキストデータを入力し、そのテキストを wiki ページに書き込むときに呼び出される。`setInput` メソッドは、コマンドなどが書かれた wiki ページ上のテキストデータを読み込み、これを PukiWikiJavaApplication (AdkService) に渡すときに呼び出される。`setSaveButtonDebugFrame` メソッドはデバッグのために使われる。`sendCommand` メソッドは PJC-S が AdkService にコマンドを渡すときに呼び出される。

#### ● Wiki ページの読み込み

PJC-S に `read` コマンドが与えられると、事前に、別のコマンドによって与えられている URL の wiki ページのソーステキストが読み込まれる。PJC-S はこのテキストを構文解析し、遠隔端末と wiki サーバ間の通信に関わるデータを抽出する。このデータは、HTML の `<pre>` と `</pre>` タグで囲まれており、遠隔端末に対するコマンドや、遠隔ページが wiki ページに書き込んだ結果などが入っている。

抽出されたデータは、`setInput(x)` メソッドを使って、AdkService に送られる。`setInput(x)` メソッドの中で、`x` から遠隔端末のコマンドが抽出され、実行される。Wiki ページ上で遠隔端末向けのコマンドは

```
[空白]command: <command>
```

の形式を持っている。

#### ● データの保存

`saveText` コマンドが PJC-S に与えられたとき、編集状態の wiki ページのソーステキストが読み込まれる。PukiWiki の場合、編集状態のページの URL は以下の形式で表すことができる。

```
[URL of the wiki page]?cmd=edit&page=[page name]
```

このページのソーステキストは、タグの対、`<form ...>` と `</form>`、を持つ。この対の中に、タグの対、`<textarea ...>` と `</textarea>`、が含まれている。`<textarea ...>` と `</textarea>` の中に含まれるテキストが、PukiWiki のページ構文で記述された PukiWiki のページのソーステキストである。従って、`<pre>` と `</pre>` タグで囲まれていたコマンドなどは、各行の最初に空白を置くことで表されている。`<textarea ...>` タグの後の行から “result:” までの行はヘッダテキストとして保存される。図 4 は編集状態の PukiWiki

ページのソーステキストの例を示す。ここで [space] は空白文字を表す。

```

<form ... > ...
<textarea ...>
...
[space]command: set data-2-2=0
[space]command: get pir stat
[space]result:
...
</textarea>
</form>
...
    
```

図 4. 編集状態の PukiWiki ページのソーステキスト

この後、PJC-S から AdkService の `getOutput()` メソッドが呼び出される。このメソッドは、コマンドの実行結果を含むテキストを返す。このテキストは PukiWiki ページ形式の整形済みテキストに変換される。変換されたテキストは保存されているヘッダテキストの後ろに結合され、あたらしいページのテキストが作られる。このテキストは、Apache の `HttpClient` の `PostMethod` によって、PukiWiki サーバに保存される。

### 3.3 AdkThread

AdkThread は Arduino からセンサデータを受け取るために使われる。センサデータの受け取りを繰り返し、受け取ったデータを AdkService に伝えるため、looping thread を持っている。現在、繰り返し頻度は 1 秒間におよそ 10 回としている。

### 3.4 AdkWikiActivity

AdkWikiActivity は、USB 接続の初期化と終了、AdkService の開始と終了、Android 端末の GUI とのやりとりなどを行う。

### 3.5 Arduino board

Arduino はセンサデータを入力し、それを Android 端末に送り、Android 端末からアクチュエータの制御データを受け取り、アクチュエータを制御する。

Arduino では、簡単な“sketch”(プログラム)が実行される。このスケッチはデータの入出力を繰り返すため、ループを持っている。このループの繰り返し頻度は 1 秒間におよそ 10 回である。Arduino と Android 端末を接続するため、USB コネクタを 2 つ持った Arduino Mega ADK を利用している。

現在、Arduino の A0 から A7 までの端子をアナログデータ入力専用端子、0 から 7 までの端子をデジタルデータ入力専用端子、8 から 13 までの端子をアナログ(PWM)/デジタルデータ出力端子としている。

図 5 に、Arduino とセンサやアクチュエータを接続した回路例を示す。

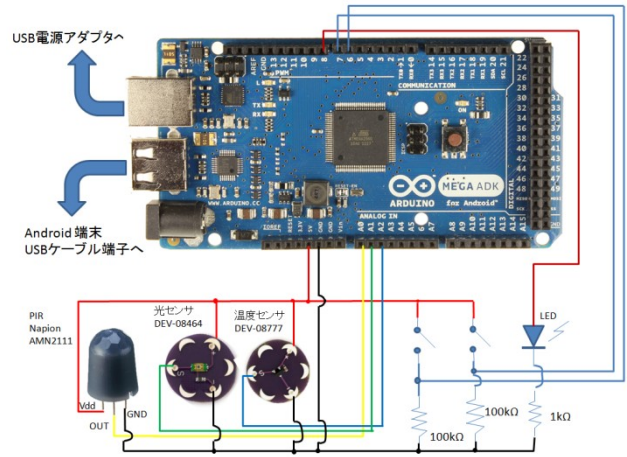


図 5. Arduino の回路例

## 4. Wiki ページに書くコマンド

遠隔端末を制御するため、Wiki ページで以下のようなコマンドを利用することができる。

- `command:set readInterval=<時間間隔>`

この Wiki ページ (コマンド) の読み込み間隔(msec)を指定する。

- `command:set sendInterval=<時間間隔>`

この Wiki ページにコマンドの実行結果を書き込む間隔(msec)を指定する。

- `command:set pageName="<ページ名>"`

次に読み込む Wiki ページを指定/変更する。このページ名の中で <hour> を入れると、この部分に現在の 24 時間表示の時間が入る。これにより、自動的に実行結果を書き込むページを時間により変更することが可能になる。

- `command:get in-a-<端子番号> <属性>`

Arduino のアナログ入力端子からデータを取得し、

`device=a-<端子番号>, Data=<日時>, <データ>`。

の形式で出力行を作り、Wiki ページの結果の最後に追加する。

<属性> は

<属性> ::= last

| stat [sampleTerm=<サンプリング間隔>]

[analysisTerm=<解析期間>]

である。last は最後にコマンドを実行したときのデータを取得し、v=<アナログ値> の形式で、出力行の<データ>の部分に埋め込む。stat は<サンプリング間隔>(msec)で示された間隔で、<解析期間>(msec)の間、データを入力し、この間に入力されたデータ列の統計をとり、以下の形式で出



力行の<データ>の部分に埋め込む。

```
ave=<平均値>, sdv=<標準偏差>,  
max=<最大値>, min=<最小値>,  
f1=<平均値からの差が小さいものの頻度>,  
f2=<平均値からの差が中くらいのものの頻度>,  
f3=<平均値からの差が大きいものの頻度>,  
n=<データの数>,  
dt=<サンプリング間隔>
```

sampleTerm=<サンプリング間隔> を省略すると、サンプリング間隔は 100msec になり、analysisTerm=<解析期間> を省略すると、解析期間は 60000 msec となる。

- command:get in-d last

Arduino のすべてのデジタル入力端子からデータを取得し、

```
device=d, Data=<日時>, v=<値>
```

の形式で出力行を作成し、Wiki ページの最後に追加する。ここで<値>は 8bit のデジタル入力端子の on/off の状態を 16 進数で表したものが入る。

- command:set out-a-<端子番号>=<値>

Arduino の<端子番号>で示す出力端子にアナログ値 (0-255) を PWM で出力することを表す。

- command:set out-d-<端子番号>=<値>

Arduino の<端子番号>で示す出力端子にデジタル値(1/0) を出力することを表す。

## 5. 利用例

われわれは本センサネットワークシステムを使って、遠隔地における人間の動作状況を把握するシステムを開発している。このシステムを使うことにより、遠隔地における人間の活動状況を入手し、その状況を Wiki ページに表示することに成功した。一ヶ月程度の期間に渡って、連続して稼働させることもできた。このシステムは人間の活動状況を把握するセンサとして PIR (焦電センサ)を利用しており、テレビカメラと比べてプライバシーへの影響が少ない。収集されるデータ量も少ないので、少ない容量の Wiki ページに長期に渡る人間の活動状況を記録することができる。このシステムを独居老人の健康確認などに利用することができる。活動状況は PukiWiki の PJC グラフ描画プラグインを使うことによりグラフで表示することができる。Wiki ページにコマンドを書くことにより、遠隔地側の LED を点灯したり消灯したりすることにも成功した。

図 6 は遠隔地に設置される遠隔端末の例を示す。この図において、PIR、照度センサ、温度センサ、LED を接続したブレッドボードが左側に表示されている。中央のボード

が Arduino で、右側の機器が Android 端末である。

図 7 に遠隔地における人間の動作状況を把握するシステムで使っている PukiWiki のページの例を示す。graph: で始まる行は、PJC により、このページのデータを使ってグラフを描画させるときの、グラフのパラメータを指定している。

## 6. 関連研究

### 6.1 Pachube

Pachube[12] は Internet of the Things のための、実時間オープンデータ web サービスの一つであり、センサデータの共有サービスの中で最も人気の高いものの一つである。Pachube はデータを upload したり操作したりするための、公開された API を持つ。このため、Pachube にデータをアップロードしたり、アップロードされたデータを操作したりするプログラムを簡単に作ることができる。

本センサネットワークシステムも Pachube と類似した昨日を持つ。しかしながら、Pachube の API は Pachube のサイトでしか使えないのに対して、われわれのシステムは、特定のサイトだけでなく、様々な PukiWiki のサイトで利用できる。

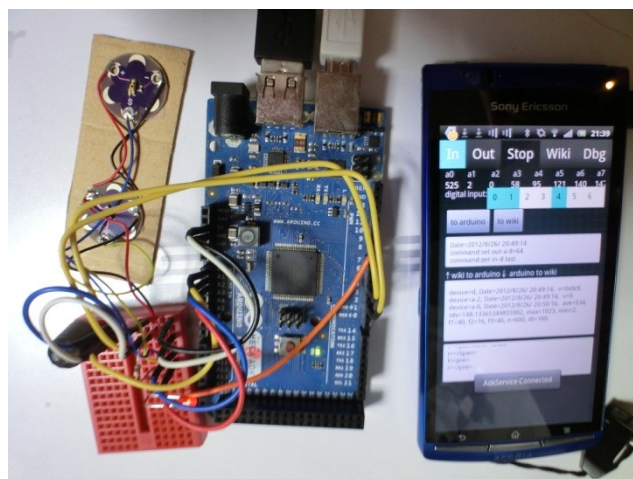


図 6 遠隔地に設置される遠隔端末の例

### 6.2 Twitter APIs

Twitter4J[15] は Java アプリケーションを Twitter と通信可能にする API である。これは Java プログラムがセンサデータを Twitter に書き込むことも可能にする。ArduinoTweetLib [10]は Arduino に Ethernet シールドをつけたものが Twitter にデータを書き込むことを可能にする API である。これらの API も Twitter との通信しか使えない。

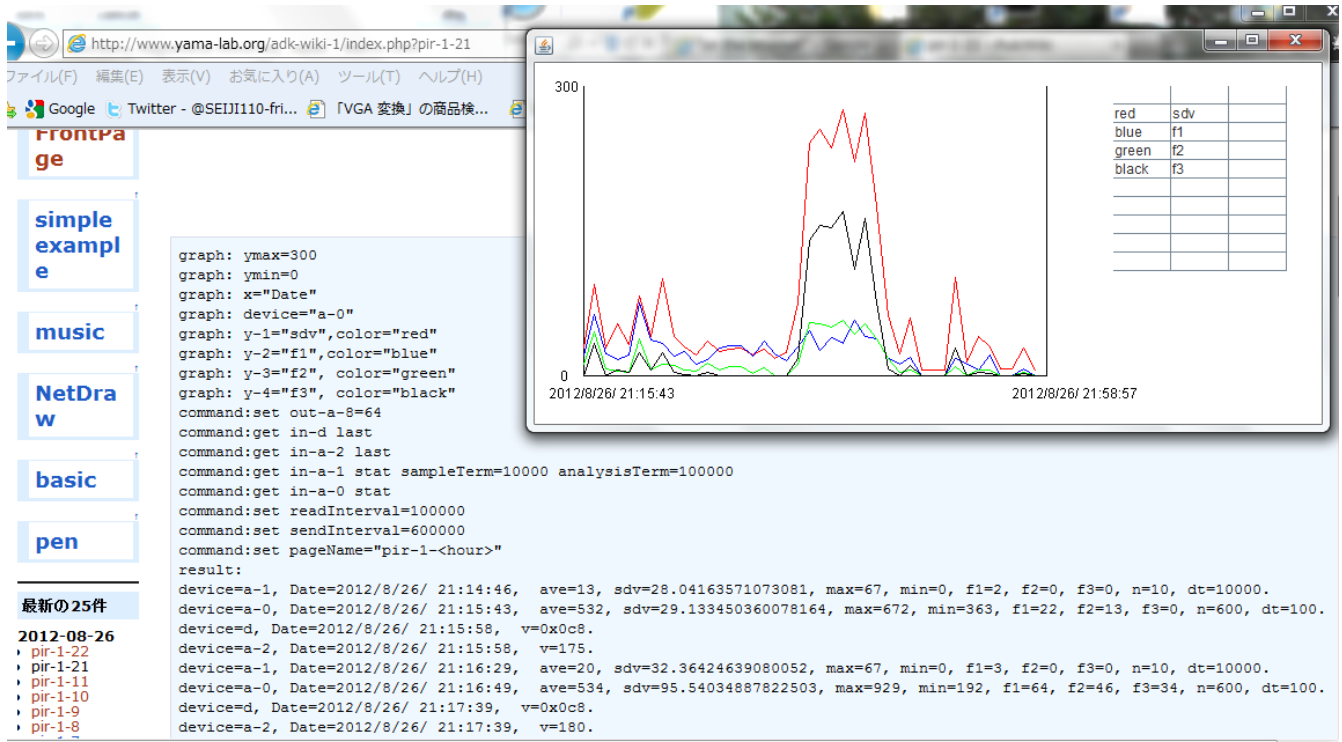


図 7. 遠隔端末を操作して、人間の活動状況等を入手する Wiki ページの例

### 6.3 TinySCADA

*Tiny Supervisory Control And Data Acquisition* (TinySCADA) system[6] は Arduino と Google App Engine (GAE) を使った M2M システムである。このシステムと本システムとは、データを入手するために Arduino を使っている点で類似しているが、TinySCADA は Android は使っていないし、GAE に依存している点で、本システムと異なっている。

### 6.4 ProtoZOA

*ProtoZOA*[1]は Arduino と ZigBee と Open Sound Control で構成されたデバイスである。ProtoZOA は Web は使っていない。

## 7. おわりに

Arduino と Android と Wiki を使ったセンサネットワークシステムと、その応用例について述べた。今後は複数のことなる Wiki ページや Wiki サーバに書き込まれたセンサデータを処理し、その結果によって、遠隔地のアクチュエータを操作するような、総合的な応用例の開発を行っている予定である。

## 参考文献

- 1) Hiroyuki Ohno: Design and Implementation of an Information Sharing Environment using Arduino, ZigBee and OSC, And Its Application to emergency Communication Systems, IPSJ Technical Report, Vol.2011-DPS-146, No.2, (2011)(in Japanese).
- 2) Takashi Yamanoue: A Draw Plug-in for a Wiki Software, saint, 10th

IEEE/IPSJ International Symposium on Applications and the Internet, pp.229-232, 2010.

- 3) Takashi Yamanoue, Kentaro Oda, Koichi Shimozono: PukiWiki-Java Connector, a Simple API for Saving Data of Java Programs on a Wiki, ACM WikiSym '11, Proceedings of the 2011 international symposium on Wikis, Mountain View, CA, USA, 3-5 oct.,(2011)
- 4) Takashi Yamanoue, Kentaro Oda, Koichi Shimozono: A Simple Application Program Interface for Saving Java Program Data on a Wiki, Advances in Software Engineering, Hindawi Publishing Corporation, 2012.
- 5) Takashi Yamanoue, Kentaro Oda and Koichi Shimozono: A Casual Network Security Using a Portable Sensor Device and Wiki Software, 12<sup>th</sup> IEEE/IPSJ International Symposium on Applications and the Internet – C3NET workshop, 2012.
- 6) Miyanishi Yohtarou: A Prototype of SCAD A System using Cloud Computing Environment GAE, IEICE technical report 111(86), 7-12, 2011-06-10(2011)(in Japanese)
- 7) Android: <http://www.android.com/>
- 8) Android Open Accessory Development Kit: <http://developer.android.com/guide/topics/usb/adk.html>
- 9) Arduino: <http://www.Arduino.cc/>
- 10) ArduinoTweetLib: <https://github.com/NeoCat/ArduinoTweetLib-server>
- 11) HttpClient Home, <http://hc.apache.org/httpcomponents-client-ga/>, 2010, As of April 4, 2011.
- 12) Pachube: [https://cosm.com/?pachube\\_redirect=true](https://cosm.com/?pachube_redirect=true)
- 13) PukiWiki: <http://pukiwiki.sourceforge.jp/>
- 14) SNORT: <http://www.snort.org/>
- 15) Twitter4J: <http://twitter4j.org/>
- 16) Wikipedia: <http://www.wikipedia.org/>
- 17) Ward Cunningham: Wiki Wiki Web, <http://c2.com/cgi/wiki?WikiWikiWeb>